FINAL PROJECT REPORT

Identification of Secret Algorithms Using Oracle Attacks

Award No. FA9550-07-1-0044

Awarded to THE RESEARCH FOUNDATION OF
STATE UNIVERSITY OF NEW YORK (CAGE code 3GRK1)
SUNY AT BINGHAMTON
4400 Vestal Pkwy E
Binghamton NY 13902-6000

Principal Investigator: Dr. Scott A. Craver
Assistant Professor, Department of Electrical and Computer Engineering
University of Binghamton
Binghamton, NY 13902-6000

Objectives

The goal of this project is the development of a system of useful tools for reverse-engineering covert channels and information hiding systems. This includes new algorithms for detection and estimation of certain hiding systems, and the statistical artifacts they leave behind. We also proposed an end-to-end system implementing our various research efforts in order to assist a specialist in breaking a covert communication system given very little information. Since it is likely for steganography to be used on very large multimedia files, e.g. audio and video, there are substantial issues to be addressed on the implementation end of such a system as well as the theoretical end.

Our project followed two tracks: as we conduct basic research in detection and estimation which comprises the primary objective of this project, we also pursued a test bed for implementing, comparing and demonstrating new algorithms. Initially we focused on audio steganalysis, but our theoretical results were generic, and for external reasons we aimed our efforts at image steganalysis.

Status of effort

Our project has focused on finding new methods to reverse-engineer detectors in short time, extending the "noise calipers" technique developed in 2006. We have applied our techniques to analyze an unknown watermark; we found it somewhat encouraging that our techniques are already well-known, and the secret watermark was specifically designed to prevent our attacks from working. Nevertheless, our analysis of modes of super-robustness led us to correctly guess much of the watermarking system's internals, towit that it used a wavelet feature space excluding the LL component.

We are now developing steganographic methods which may be immune to statistical steganalysis, by embedding data in high-level content of a statistically artificial videoconferencing channel. This "supraliminal" channel, as it is called in the literature, attempts to circumvent normal methods of statistical steganalysis by avoiding the strategy of embedding data in conventional multimedia data. Instead, data is embedded in computer animations, which are now usable as backdrops in popular videoconferencing software.

Accomplishments/New Findings

The Noise Calipers Technique

Suppose that we have a watermark detector, any generic detector that we want to reverse-engineer. We can attempt to submit experimental images, whose output will help us deduce the algorithm's inner workings. This *operational information leakage* is difficult to avoid, even if the algorithm itself can be kept secret.

In a more recent challenge, the PI and his students had three months to reverse-engineer and break an image watermarking system. On the right is one of the watermarked images, superimposed with an experimental image. This attack exploited what we now call *super-robustness* of watermarking systems. Watermark detectors sometimes admit extreme false positives, which leak information about the algorithm. Such a severe change as illustrated should break a watermark, but it won't break watermarks that are embedded in 8-by-8 pixel blocks. Hence the mark's survival tells us about the detector.

Extending these results, we have designed general techniques to force a watermark detector to leak specific information about its secret algorithm. If the watermark uses normalized correlation in its

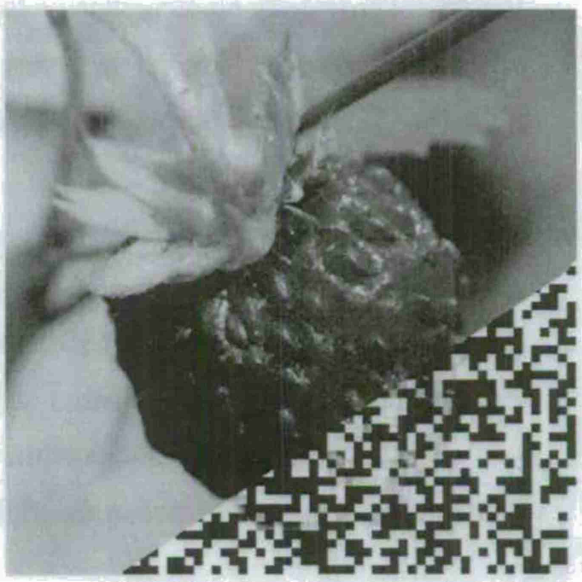


Figure 1: A challenge image from the BOWS contest, superimposed with one of our experimental attacks.

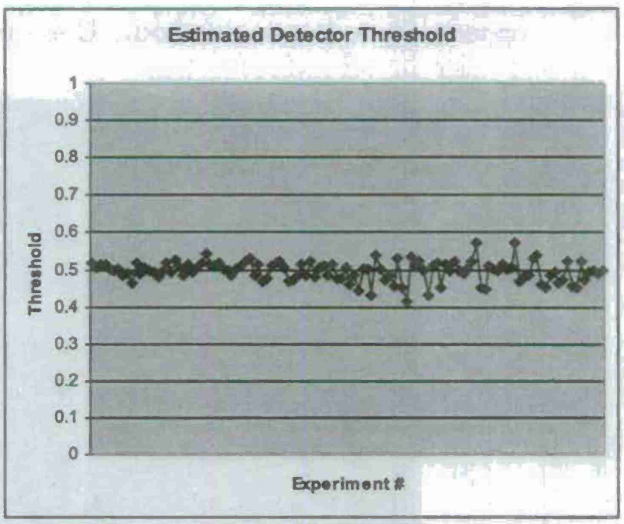


Figure 2: a detection threshold of 0.5, estimated by an average of 1016 detector queries per experiment. With 500 detector features, this detector has an asymptotic false alarm rate of 2.39×10^{-33}.

detection, we can deduce parameters such as the number of watermark features and the watermark detector threshold. In an interesting experiment, we were able to estimate the false alarm rate of a detector by querying it 1,000 times—even though the false alarm rate was on the order of 10^{-33}. This exploits the same super-robustness principle: we iteratively grow long noise vectors under which the watermark remains detectable, and when they grow to sufficient length they tell us properties of the detector, such as a normalized correlation threshold. A similar experiment tells us the number of features used.

Unfortunately, not all detectors can be polled indefinitely to leak information about their inner workings. In some scenarios, we have the opportunity to submit a small set of inputs, e.g. on the order of 10-100 inputs. Thus we need fast techniques to reverse-engineer an unknown detector based on few experimental interactions.

The technical details of the BOWS contest and the Noise Calipers techniques are now published in the EURASIP journal of information security.

Application of Superrobustness modes in BOWS II

A second watermarking contest, BOWS II, has provided more watermarked images to reverse-engineer. The secret algorithm was not revealed until 2008, giving us the opportunity to test our methods. One example attack is shown below:

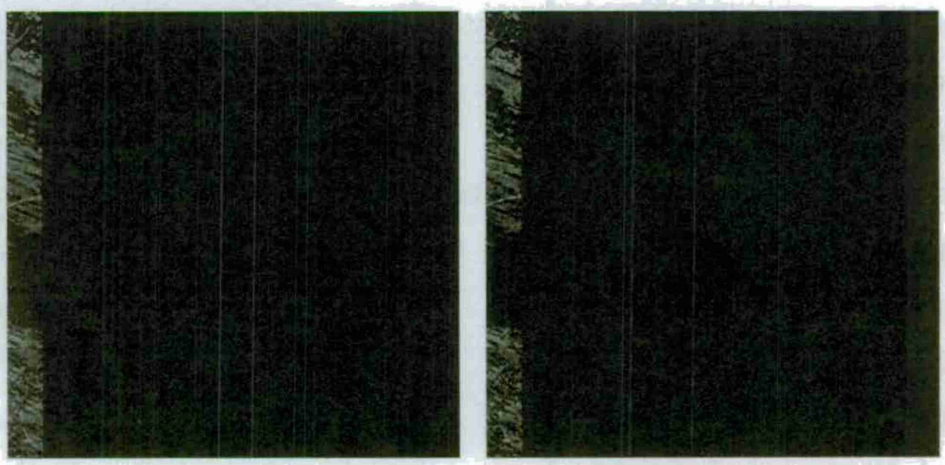


Figure 3: a watermark survives when an image is severely cropped, but detection fails if the cropped region is given some energy.

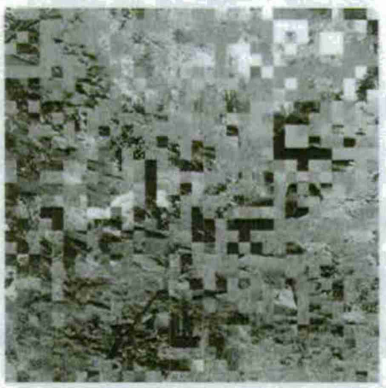
An image, when cropped to the leftmost 40 pixels, passes the detector; that is, the watermark survives. Yet slight random noise injected into the cropped space causes a detector failure. This is strong evidence that the detector uses some form of normalization, for example extracting image features and then performing normalized correlation with a target watermark vector. In such a detector, an extracted feature vector x is compared to a watermark w using a formula like $f(x) = x \cdot w / ||x||$. If the image is cropped so that only some fraction α of the vector remains, the detector statistic becomes $f(x|_\alpha) \approx \alpha x \cdot w / \sqrt{(\alpha \sum x_k^2)} = \sqrt{\alpha}\, f(x)$. This represents the image on the right, where most of the image is removed.

If on the other hand the removed data is replaced with a random signal z of energy $\beta||x||$, the statistic becomes $f(x|_\alpha + z) \approx (\alpha x \cdot w + z \cdot w) / \sqrt{(\alpha \sum x_k^2 + \sum z_k^2)} \approx f(x)\alpha / \sqrt{(\alpha + \beta^2)}$. This represents the image on the right: the crucial difference is the presence of an extra factor β^2 in the denominator, making the statistic smaller. In other words, adding random noise does not change the $x \cdot w$ part of $f(x) = x \cdot w / ||x||$, but it increases the magnitude $||x||$, reducing the watermark strength. The manipulation of parameters α and β can be used to identify particular types of watermark detection algorithms.

Combination of multiple watermarked images have some usefulness in reverse engineering, but our experiments show them to be limited in their selectivity. We observe that multiple images have the same watermark, and submitting any of three images to the same detector yield a positive result. Expanding on this, we combined the images in various ways. First, we submitted weighted averages of our images to the watermark detector, finding that these are always detected as watermarked. This is illustrated in figure 4.

Figure 4: all weighted sums of images has a recognizable watermark.

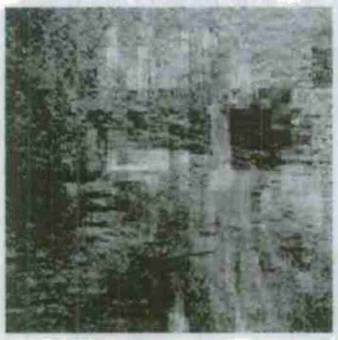
Next, we created "patchwork" images by assembling parts of the three images at random. The detection depended on the block size. An 8x8 patch renders the mark undetectable, indicating that the detection does not use 8x8 blocks; however, above that the watermark is detectable.



Block Size	Detection
8	No
16	No
32	Yes
64	Yes

Figure 5: a patchwork image, and results from patchwork images of varying patch sizes.

We attempted to use this technique in a more general algorithm. If we could guess the feature space used by a detector, that should survive any patchwork constructed from multiple images. Thus we attempted patchworks of varying block sizes in different watermark feature domains, such as the Haar wavelet and DCT domain. Results show that regardless of block size or choice of domain, the watermark remained detectable. This did not give us a test with useful selectivity between domains. Instead, it seems to show that the failure of patchwork in the spatial domain is an anomaly.



Domain	Block	Detection
Haar	8	Yes
Haar	4	Yes
Haar	2	Yes
Haar	1	Yes
CDF	8	Yes
CDF	4	Yes
CDF	2	Yes
CDF	1	Yes

Domain	Block	Detection
DCT	16	Yes
DCT	8	Yes
DCT	4	Yes
DCT	2	Yes
DCT	1	Yes

Figure 6: patchwork attacks in different embedding domains.

Results

The algorithm for BOWS-II was eventually published, and its description is available on the BOWS-II web site. The "Broken Arrows" algorithm, designed by Teddy Furon and Patrick Bas, does use a wavelet transform, in particular a Daubechies 9/7 Wavelet, in all but the LL subband. The watermark was not embedded strictly additively, but in such a way as to maximize the minimum distance from the watermarked image to the detection boundary.

We were struck by the fact that the algorithm was specifically designed to halt our own specific attack methods. The detection boundary was purposefully made ragged so that our growth of "noise snakes" would be hindered. The detection region was also bounded, preventing certain modes of superrobustness from being identified.

Identification of +/- K embedding

As separate track from reverse-engineering watermark algorithms, the PI developed methods to better detect and estimate +/-K embedding, a common form of steganography. This research project was undertaken at the Air Force research lab in Rome, NY, under the mentorship of Chad Heitzenrater, AFRL/IFEC.

In +/- K embedding, a message is embedded in an image by either incrementing or decrementing the luminance value of each pixel by a fixed value K. The data is encoded in the sign of the luminance change, and can be concealed by using a small fraction of pixels or weak embedding constant. The value of K and embedding rate are both important parameters that we wish to estimate.

Our technique is to observe that additive noise signals induce a convolution in the intensity histogram of an image. If we denote h_x as our image histogram and p_w as the probability distribution of our additive watermark, the marked image has an intensity histogram $h_y = h_x * p_w$. This implies a multiplicative relationship in the Fourier domain, and an additive relationship in the log-spectral domain: $ln\ F\{h_y\} = ln\ F\{h_x\} + ln\ F\{p_w\}$. Alternately we can work in the cepstral domain: $F\{ ln\ F\{h_y\}\} = F\{ ln\ F\{h_x\}\} + F\{ ln\ F\{p_w\}\}$.

This suggests a technique for estimating the distribution of an added watermark: compute the log spectrum of an image's intensity histogram, then run it against a bank of correlators for common spectral signatures $ln\ F\{p_w\}$. This requires brute force over different parameters, but in +/-K embedding, there are not many choices for K.

We refined this technique somewhat, by computing cepstral signatures for separate patches of the image, reasoning that distinct regions of the image may be statistically different from one another; and by using different domains from the pixel domain. In particular, we achieved useful results by replacing the histogram of pixel intensities with histograms of pixel *differences:* we take each pixel value minus that of its immediate right neighbor. This has a more well-behaved distribution, and +/- K embedding still induces a filtering effect in this domain.

The figures below illustrate the progressive refinement of our detector, and the reduction of noise in the estimation of embedding parameters.

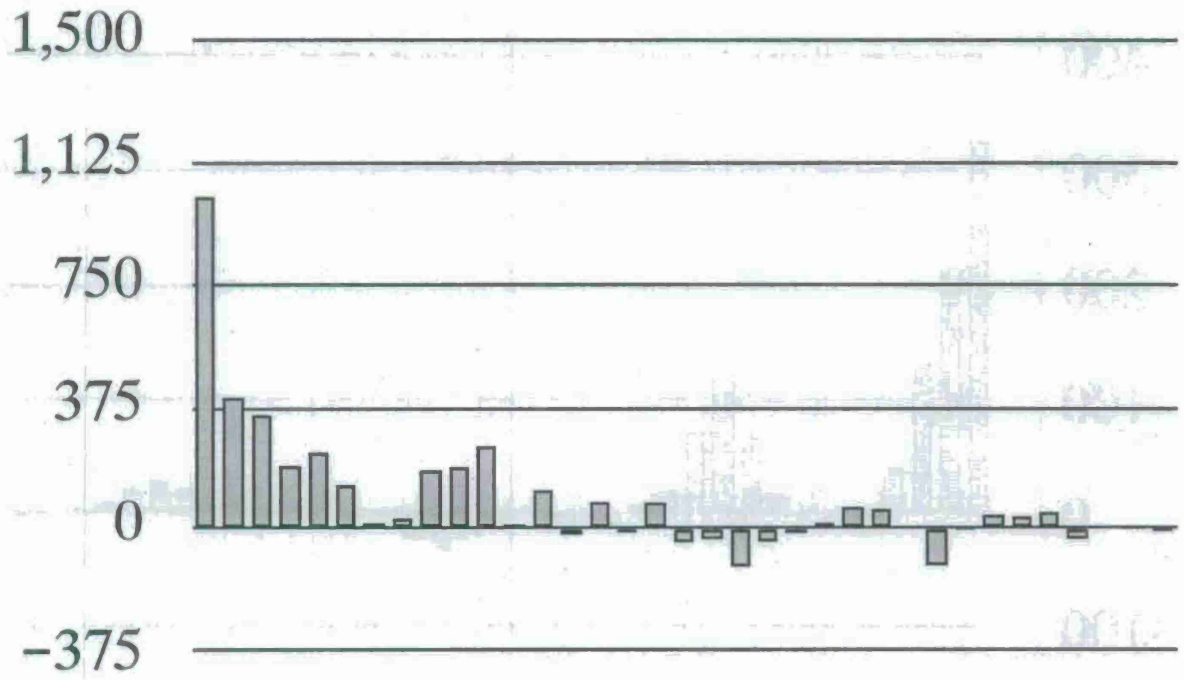


Figure 7a: histocepstrum of an image with a watermark of magnitude +/-5.

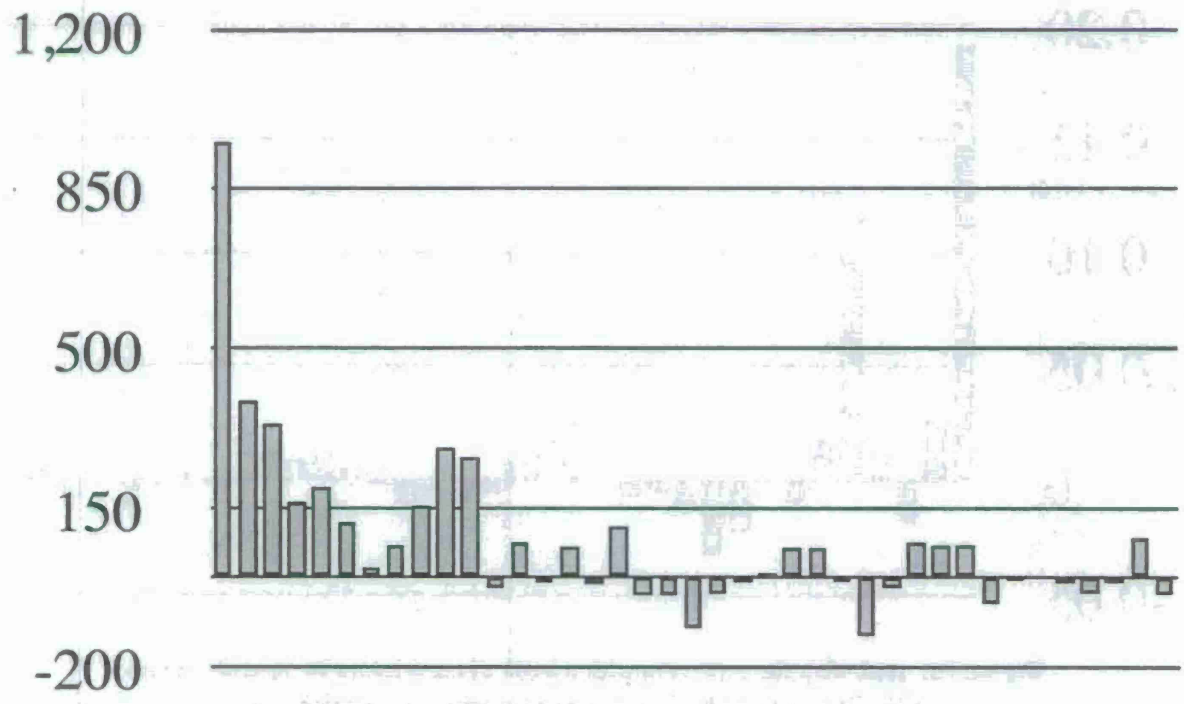


Figure 7b: "folded" histocepstrum with positive and negative frequencies combined.

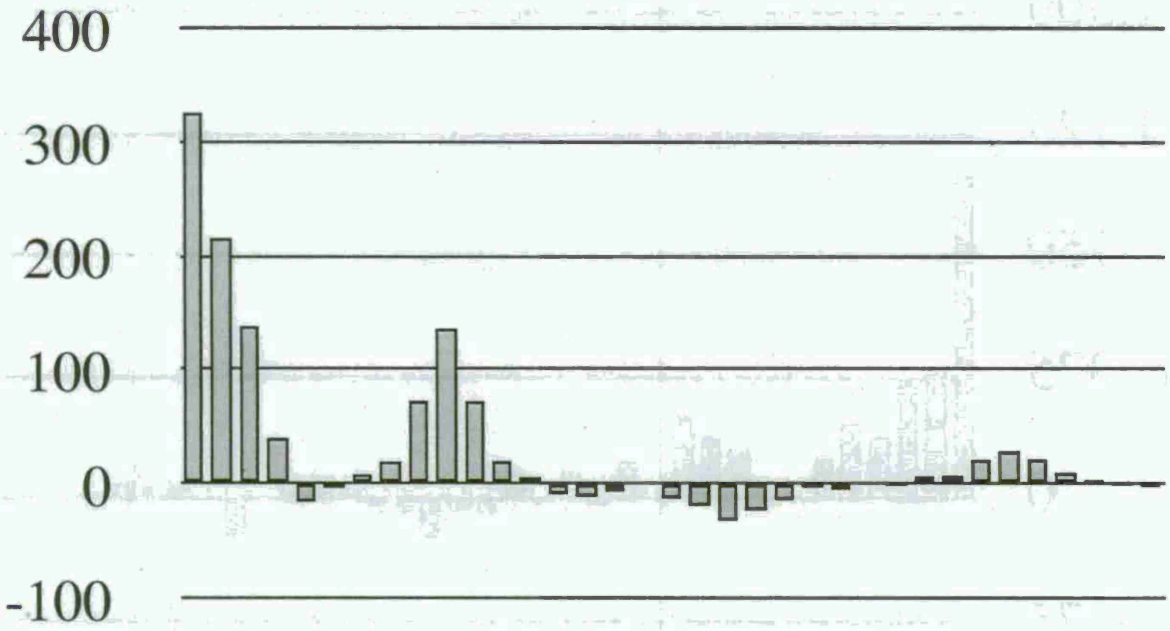


Figure 7c: "folded" histocepstrum averaged over multiple 128x128 blocks of the image. The effect is much clearer when each block is analyzed separately.

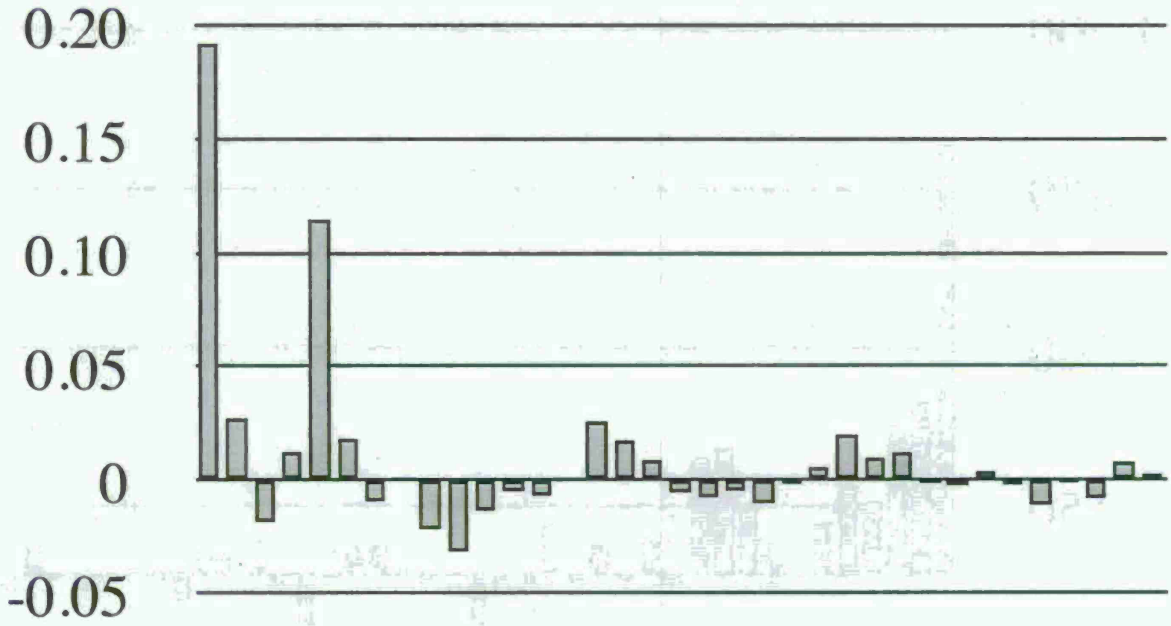


Figure 7d: replacing the cepstrum with a bank of correlators for specific values of K. The spike at K=5 is now clearer, with weaker sidelobes.

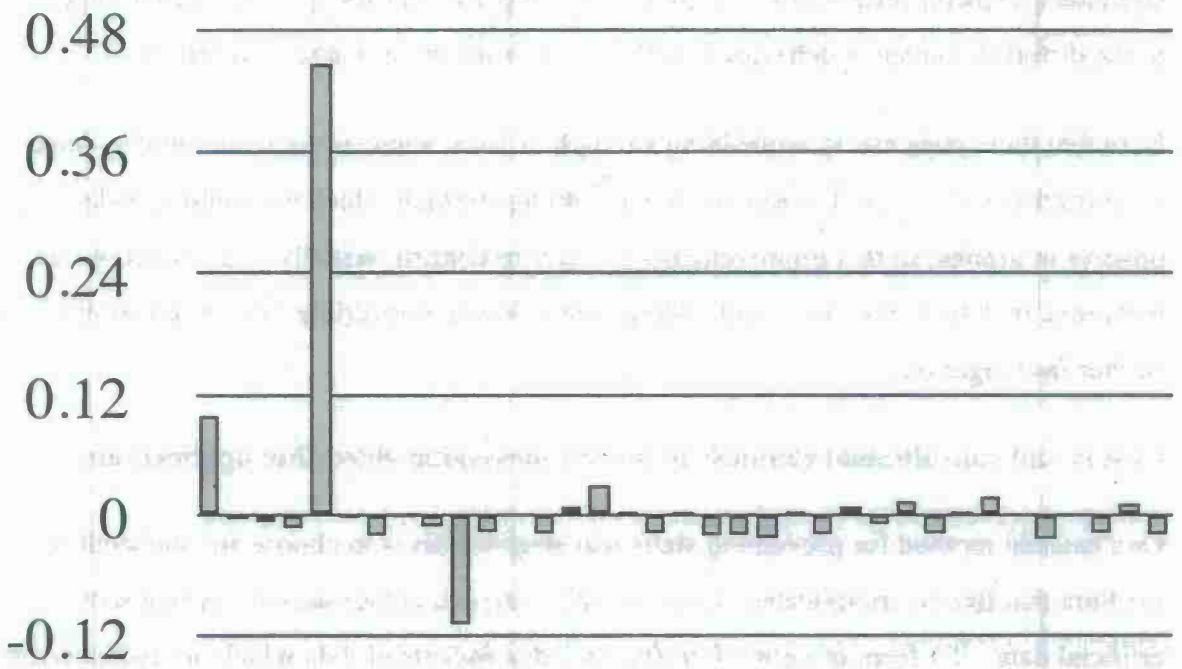


Figure 7e: detection using histograms of adjacent pixel differences.

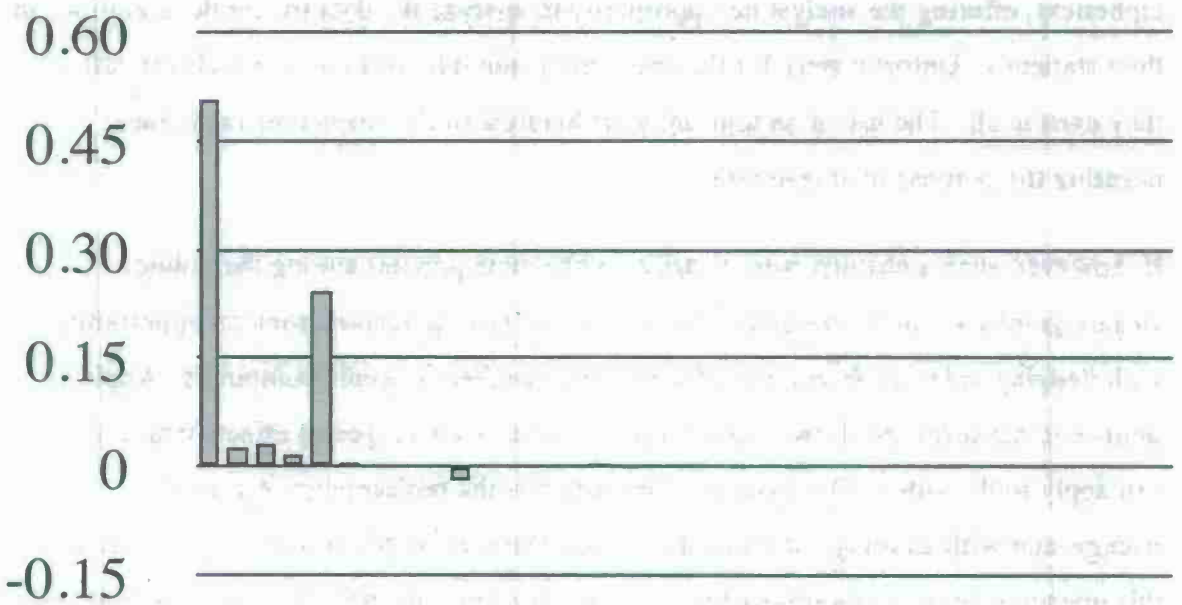


Figure 7f: detection at 25% embedding.

The main useful result is that additive watermarks in some domain may be more easily detected if we search for cepstral signatures not in the domain itself, but in the adjacent differences of pixels or other features. Working in a patchwork style, for example

combining cepstral results from different image regions, can also provide resolution due to the different statistical behavior in different portions of an image or spectrum.

Note that these tests use an embedding strength of K=5, whereas we would like to detect watermarks at +/- 1. We focused on an embedding strength which we could visually observe in graphs, so that improvements are easy to confirm visually. The effect of this technique in detection of +/- 1 embedding, and at lower embedding rates, is a matter of further investigation.

Covert and supraliminal channels in instant messaging video chat applications

One unusual method for preventing statistical steganalysis is to choose an emebedding medium that has no complicated statistical behavior, and is thus easy to replace with artificial data. If a form of network traffic includes packets of data which are independent and uniformly distributed, for example, those packets could easily be replaced with ciphertext, offering the analyst no opportunity to analyze the data for subtle alterations in their statistics. Unfortunately for the transmitter, suitably artificial channels are rare, if they exist at all. The use of an unusually artificial carrier is suspicious on its face, negating the purpose of steganography.

If, however, such a channel were to arise and become popular among the public, steganography would be possible. We believe we have perceived such an opportunity with fledging videoconferencing software now installed in Apple computers. Apple's built-in iChat software allows videoconferencing as well as special effects which a user can apply to the video. One type of visual effect is the replacement of a user's background with an image or video file. Upon learning of this feature, we realized that this opened a unique opportunity for a communications channel: Apple QuickTime, the native wrapper for video files, can also contain computer animations, and with some tampering an animation can base its display on the value of external ciphertext instead of a pseudo-random number generator.

We suspected that if a video file could be placed in the backdrop of a video chat session, so could a computer animation modulated by ciphertext. This is an example of a

supraliminal channel: a channel in which data is represented as meaningful semantic content, which cannot be removed by adding noise. Rather than concealing the data, it is overtly represented and can be decoded by anyone. To achieve security, the channel must be designed so that innocent data also decodes to random bits which resemble an encrypted message.

To test this theory we created a computer animation, and a custom animation component (these can be written in C, and stored in a dynamic library that is linked into the QuickTime library.) Our component, or *patch*, acts as a random number generator commonly used in computer animations---except that it connects to a server to request the random data. This server, written in Tcl with C extensions, can be given message data, and gives the patch either encrypted message data or pseudo-random bits if no message data is present. Hence the random input to the animation is modulated with message bits. The final component of this system is a program which can analyze the video from a video chat system and extract the message bits.

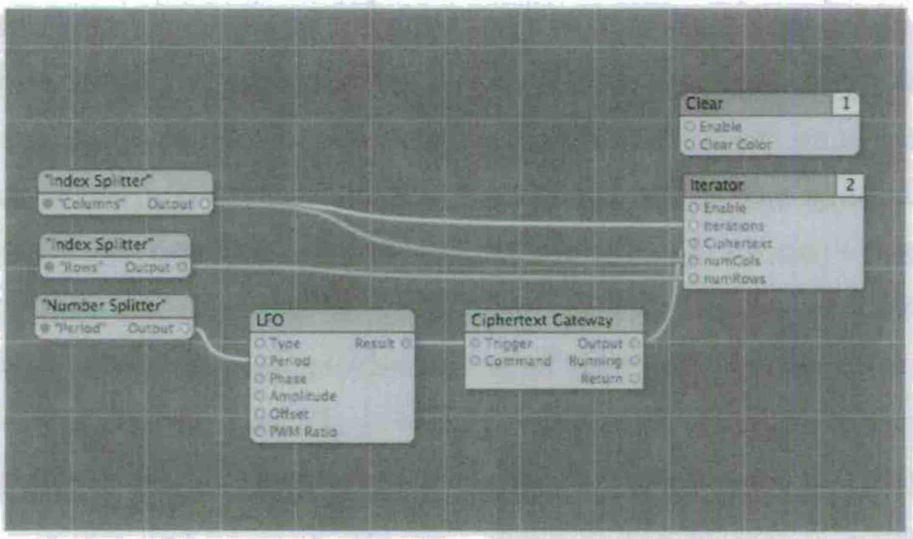


Figure 8: A graph of patches describing a Quartz Composer animation. Our custom ciphertext gateway patch (bottom right, with square corners) masquerades as a legitimate system component, and imports ciphertext into the animation's pseudo-random data stream.

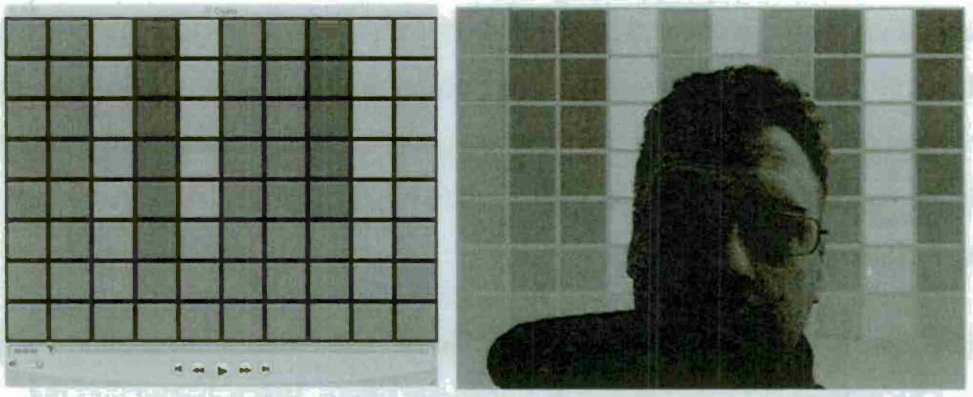


Figure 9: Left, the animation generated from covertly modulated pseudo-random data. At this instant it encodes the octal number 7354006242 (hex 0x3bb00ca2)---the hue of each column is based on one octal digit. Right, the animation as an iChat backdrop.

In our explorations, we found that the QuickTime video wrapper that embedded computer animations initially refused to incorporate our ciphertext gateway; for security reasons, encapsulated video files cannot connect to the Internet, access certain files, or sample devices such as the microphone or camera. Nor can encapsulated video files include third-party components; only animation components included in Apple's Quartz Composer framework can be used, and only those that are considered safe. However, this was trivial to circumvent. No code-signing is used to mark safe components, and examination of the binary files revealed that Apple components simply implement a Boolean `isSafe` method: the Quartz Composer patch object has an undocumented class method `+(BOOL) QCPatch isSafe` which defaults to `NO`. We simply included the method descriptor in our code, subclassed `QCPatch`, and overrode `isSafe` to output `YES`. This weak form of sandboxing code identifies a security risk: if a corrupt patch is placed in the appropriate directory of a user's account, any QuickTime video file the user plays or watches on the Internet could access the user's file system and covertly upload or download information.

This allows a form of data embedding by shaping the actual content being generated, rather than hiding data in already-generated content. The two main goals are robustness against an adversary (who may be allowed to add noise, but not change semantic content of a message) and plausible deniability. If the use of videoconferencing backdrops becomes sufficiently popular, those which are modulated by random data can be turned

into covert channels by seeding the random generator with ciphertext; assuming that the random generator and ciphertext possess computationally indistinguishable outputs, such a method allows a means to circumvent statistical steganalysis.

Random Dot Watermarking

It is well known in digital watermarking that an adversary can reverse-engineer a watermark detector and often uncover a secret watermark with a large number of experimental inputs to the detector. These so-called sensitivity attacks are able to defeat many detectors with an amount of effort proportional to the dimension of a watermark feature space. In the extreme case, a simple watermark correlator can be reverse-engineered by acquiring n points on the boundary of the detection region, and solving the n-dimensional equation for the planar surface of the detection boundary.

In cryptography, it is typical for an n-bit secret key to require effort proportional to 2^n (or at least effort that is superpolynomial in n) to reverse-engineer a key. That detectors only require polynomial effort, and often linear effort, is a surprising deficiency of watermarking systems.

We have developed a technique for constructing a randomized watermark detection algorithm that requires exponential rather than polynomial effort. This may be the first watermarking algorithm designed to be systematically resistant to sensitivity attacks, impeding reverse engineering by design rather than by ad-hoc measures, and providing a substantial asymptotic increase of attack effort.

This technique, called random dot watermarking, replaces a customary correlation detector with a large pseudo-random family of correlator detectors, each with a very high threshold. On the signal sphere, a customary correlation detector can be seen as an almost hemispherical detection region; the random dot approach can be seen as a union of many very small circles, positioned in random locations. The dots are large enough to allow robust watermark embedding, numerous enough that most signals are close to a dot and therefore easy to watermark, and small enough that the detector has a small false alarm rate. The effect is that a single compute image is watermarked by moving it into one random dot; this dot's location is statistically independent of the remaining dots, so a sensitivity attack does not yield any useful information for attacking any other image.

Coin Flip Channels

Earlier in this project we developed methods for secure communication through supraliminal channels in videoconferencing sessions. For this, we concealed ciphertext by using it as pseudo-random input to a computer animation, which was then captured by a recipient and decoded to the original value. Subsequently, we established similar channels in smart-phone walkie-talkie applications.

Figure 1: screenshots of an iPhone walkie-talkie application that embeds secret text in ambient sound effects.

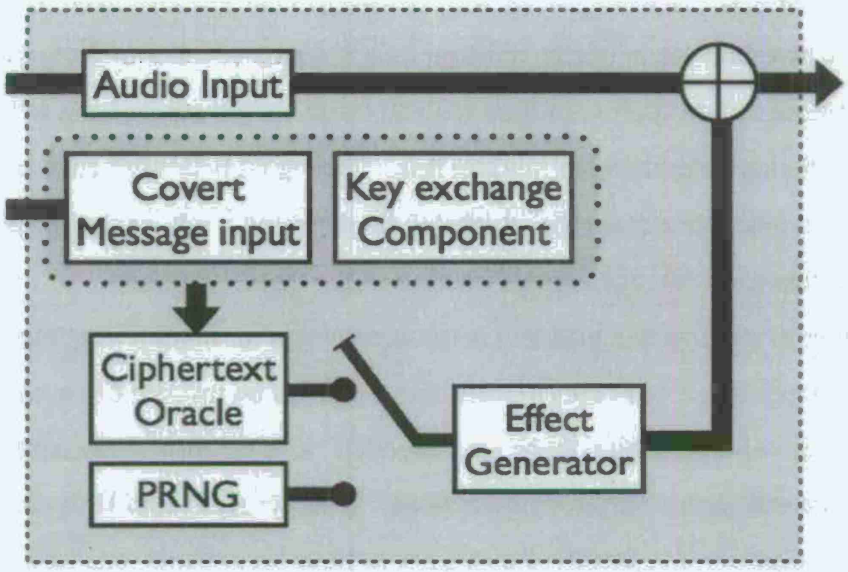


Figure 2: The block diagram for embedding. This is similar in design to our videoconferencing application data hiding: message data is encrypted and used in place of PRNG data to seed an effect generator.

A major problem with this type of channel is that the embedded data must not exhibit any structure identifying it as a message; it must look like random noise, and any embedded cryptography must be indistinguishable from the random noise that it replaces. In some scenarios this is easily obtained. For example, if Alice and Bob communicate with a secret key, they can transmit data that has been XORed with a cryptographically secure pseudo-random key stream. This produces data that is indistinguishable from random coin flips, and can also be immune to noise in the channel. The underlying data can be protected by an error correcting code, and bit errors inflicted upon the ciphertext are translated to identical bit errors in the plaintext.

However, if Alice and Bob do not share a secret key in advance, can they use this channel to perform key exchange? We found that the answer is yes, if no noise exists on the channel, but no if an adversary can flip even a vanishingly small fraction of bits. This unexpected result tells us that in some circumstances even provable steganography can be defeated by even a very slight active warden.

In our analysis, we modeled these problems as coin flip channels. In a coin flip channel, Alice and Bob are both transmitting long streams of iid coin flips, and are allowed to replace any of their coin flips with any data they want, as long as the result is statistically indistinguishable from random bits. Their goal is to send each other fake randomness that encodes a message, ultimately performing a key exchange protocol. An adversary is allowed to corrupt a small fraction of the underlying bits.

The key to solving this problem is to observe that no matter what methods Alice and Bob use to get data across the channel, it can always be modeled as a code: Alice transmits a large number of bits, these are "decoded" as a smaller string, and thus the string was represented as a very long codeword. Whether Alice and Bob use a codebook or an elaborate algorithm to immerse messages in their bit streams, one can always characterize the messages in this way. Then, we exploit a fact from high-dimensional geometry: if we consider all the codewords corresponding to a string s and choose a word uniformly from this set, it is almost always a few bits away from a non-codeword. This is due to the fact that in high dimensions the grand majority of a hypersphere's interior is within a small distance of its surface. The import of this is that a uniformly chosen codeword is only a few bit errors away from being misdecoded. An adversary who knows the protocol can force decoding to fail with very little effort.

Personnel Supported

Scott A. Craver, principal investigator.
Assistant Professor, Department of Electrical and Computer Engineering
University of Binghamton
Idris Atakli, graduate student
Jun Yu, graduate student.
Enping Li, graduate student.

Publications:

This effort resulted in several new discoveries in data hiding, notably embedding in pseudo-random state (coin flip) channels, a refinement of noise caliper techniques, and the development of watermark algorithms immune to reverse-engineering. The latter

discovery is a recent one and as yet has not been submitted for publication. The previous discoveries are enumerated in the following journal article and conference papers. We expect several more paper submissions to stem from this research effort, as the graduate students supported by this award complete their dissertations.

1. **E. Li and S.A. Craver**, "A Supraliminal Channel in a Wireless Phone Application." in 11th proc. ACM Multimedia and Security Workshop, NJ USA, Sept 7-8 2009. pp 151-154.
2. **S.A. Craver, E. Li and J. Yu**, "Protocols for Data Hiding in Pseudo Random State." in Proc. SPIE, Media Forensics and Security, Jan 19 2009, San Jose, CA. Vol. 7254, 72540W.
3. **J. Yu and S. A. Craver,** "Reverse-engineering a watermark detector based on a more precise model" Submitted to SPIE, Media Forensics and Security 2010
4. **S.A. Craver, I.M. Atakli, and J. Yu**, "Reverse-Engineering a Watermark Detector Using an Oracle" EURASIP Journal on Information Security, vol. 2007, Article ID 43034, 7 pages, 2007. doi:10.1155/2007/43034.
5. **S. Craver, E. Li, J. Yu, and I. Atakli.** "A supraliminal channel in a videoconferencing application." *Information Hiding*: 10th International Workshop, IH 2008, Santa Barbara, CA, USA, May 19-21, 2008. Revised Selected Papers. LNCS vol 5284/2008, pp 283–293.
6. **S. Craver, E. Li, J. Yu, and I. Atakli.** A supraliminal channel in a videoconferencing application. In Information Hiding: 10th International Workshop, IH 2008, Santa Barbara, CA, USA, May19-21, 2008, pages283–293.
7. **S.A. Craver, J. Yu and I. Atakli,** "How We Broke the BOWS Watermark." In Proceedings of The International Society for Optical Engineering, E. J. Delp III and P. W. Wong, Eds., vol. 6505 of Proceedings of SPIE, San Jose, Calif, USA, January 2007.
8. **S.A. Craver and J. Yu,** "Reverse-engineering a Watermark with False Alarms." To appear, ." In Proceedings of SPIE, Security and Watermarking of Multimedia Contents IX, January 2007.

Interactions/Transactions:

We have provided consultative and advisory functions to AFRL in Rome, NY. As part of a summer faculty program the PI has performed work in image steganalysis relevant to this award, as described in section "Identification of +/-K embedding."

New inventions or patent disclosures:

There have been no invention or patent disclosures.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
28-01-2011	Final Performance	01-01-2007 to 05-31-2010

4. TITLE AND SUBTITLE
Identification of Secret Algorithms Using Oracle Attacks

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-07-1-0044

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Scott A. Craver

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
STATE UNIVERSITY OF NEW YORK (CAGE code 3GRK1)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
875 N. Randolph St
Suite 325, Room 4036
Arlington, VA 22203-1768

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0081

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A - Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The goal of this project is the development of a system of useful tools for reverseengineering covert channels and information hiding systems. This includes new algorithms for detection and estimation of certain hiding systems, and the statistical artifacts they leave behind. We also proposed an end-to-end system implementing our various research efforts in order to assist a specialist in breaking a covert communication system given very little information. Since it is likely for steganography to be used on very large multimedia files, e.g. audio and video, there are substantial issues to be addressed on the implementation end of such a system as well as the theoretical end.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	U		

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.